IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	: 10/733,469
Applicant	: Victor J. YODAIKEN
Filed	: December 12, 2003
TC/A.U.	: 2135
Examiner	: Suman Debnath
Docket No.	: 0125-143
Customer No.	: 06449
Confirmation No.	: 8829

DECLARATON OF VICTOR J. YODAIKEN PURSUANT TO 37 C.F.R. § 1.132

I, VICTOR J. YODAIKEN, hereby declare and state as follows:

I. Background and Experience

1. I founded Finite State Machine Labs, Inc. ("FSMLabs"), the assignee of U.S. Patent Application No. 10/733,469, and am currently the President of FSMLabs.

2. I hold a Ph.D. in Computer Science from the University of Massachusetts and a Master of Science in Computer Science from the University of South Carolina. I have worked in the systems software industry for over 26 years. I am well-known in the community of hard real-time operating systems as the inventor of *RTLinux* (Real-Time Linux).

3. I consider myself to be at least one having ordinary skill in the art in the field of hard real-time operating systems to which the claims of the present application are directed.

4. I reviewed the present application, the Office Action dated December 14, 2007, and cited prior art references (U.S. Patent Nos. 7,152,242 ("Douglas"), 6,640,242 ("O'Neal"), and the cited portion of the website for REDSonic, Inc. ("REDSonic")).

II. The term Real-Time

5. The term "real-time" is an oft misunderstood and misused term. The term real-time is most often used to refer to something called "soft real-time," which relates to applications that require some speed yet can tolerate latency.

6. The term "real-time" does not have the same meaning as the claim term "hard real-time."

II. Hard Real-Time Operating Systems

7. Hard real-time operating systems are operating systems intended for use in applications that require exact response time and cannot tolerate latency (i.e., hard real-time applications). Such applications include, for example, embedded systems for various electronic devices, industrial robots, spacecraft, industrial control, and scientific research equipment. Typically, a hard real-time operating system performs tasks within a very-small, specified, fixed period of time. For example, the present application discloses a system that can guarantee a response in as small a time frame as 1 millisecond (ms). See, ¶ [0014] of the present application.

8. An exemplary hard real-time operating system includes *RTLinuxPro*. (See e.g., ¶ [0024] of the present application).

9. Before the time of the present invention, hard real-time operating systems existed, however they had not been used to provide security features for other software systems. (See e.g., ¶ [0006] of the present application).

III. The Claimed Invention

10. Traditional security techniques for computer software include passwords and other authorization tokens, use of encryption, and permission checking systems. In such standard methods, "security markers" (e.g., checksums, digital signatures, and permission levels) and "security properties" (e.g., an exact match between a data item that is supposed to be immutable and a hidden copy of that data item) can be used to validate the integrity of data and of the security system. (See e.g., ¶ [0002] of the present application).

11. The present application describes systems and methods for applying hard real-time capabilities to software security. The claims describe several embodiments which take advantage of the unique properties of hard real-time operating systems to provide security features not previously available using traditional methods. The hard real-time features of the security process are important where executing application code periodically performs an important task and the consequences of it not performing this important task in an intended manner could be dire. Thus, the security process, which has the ability to shut down code before it is scheduled to perform an important task when there is an indication that code has been tampered with, is highly valued. (See e.g., ¶ [0031] of the present application).

12. A hard time limit can be used for security processes which cannot be achieved with regular operating systems. (See e.g., ¶ [0035] of the present application).

13. One skilled in the art reviewing the present application would understand that the systems and methods for detecting a security breach disclosed are unique to

hard real-time operating systems and could not be accomplished using traditional security techniques.

IV. The Prior Art Cited in the Office Action

14. As stated above, I have reviewed the Douglas, O'Neal, and REDSonic references. In my opinion, one skilled in the art at the time that the present invention was filed would not have consulted the Douglas and O'Neal references because they do not relate to hard real-time operating systems. Further, as I explain in further detail below, none of the cited prior art actually discloses security features using hard real-time operating systems that would have been understood by those skilled in the art.

A. The Douglas Patent

15. The Abstract for Douglas describes an "intrusion detection" system that continuously monitors a computer system for security compromise stating:

> [a] host-based intrusion detection system (HIDS) sensor that monitors system logs for evidence of malicious or suspicious application activity running in real time and monitors key system files for evidence of tampering.

(Abstract). Douglas uses the term "real-time" here as it is commonly used in the intrusion detection literature to mean "on-line" or "continuously updating." The term of art "hard real-time" is often used in a completely different sense to refer to systems that are characterized by time constraints and/or deadlines and/or deterministic execution. The term "hard real-time" always refers to time constrained systems that have particularly stringent timing requirements and is never used in the sense of "on-line". In rejecting claim 1, the Examiner cites the Abstract of Douglas as disclosing "a computer executing a hard real-time operating system." Douglas never uses the term "hard real-

time" in the Abstract or anywhere else in the patent. Moreover, Douglas does not contemplate any of the concepts of hard real-time, including deadlines, deterministic execution, operating systems that can guarantee timing (hard real-time operating systems), or any of the associated technologies or methods of hard real-time systems. (See e.g., ¶ [0031] of the present application).

16. Here Douglas is clearly using "realtime" to mean "ongoing" and not in any sense to refer to timing constraints. For example, Douglas states:

> [o]n Unix platforms, the HIDS sensor 20 can be configured to display events to the screen as they occur. This feature can be useful when testing/debugging to watch events in realtime.

(Col. 11, Lns. 19-21). The Abstract of Douglas further refers to examination of log files that are produced as applications and systems run, with no timing constraints at all noting:

> [t]he system monitors the logs of applications running on the host, including mail servers, web servers and FTP servers. The system also monitors system files and notifies the system administrator when key system and security files have been accessed, modified or even deleted.

(Abstract). It is clear that Douglas refers to the term "real-time" as it is employed in the standard literature of intrusion detection monitoring and not in the sense of "hard real-time."

17. The use of the term "real-time" as meaning "on-line" rather than hard real-time operating systems as in the present application was well known at the time of the invention. For example:

> Although often ignored in off-line analysis, efficiency is a very important consideration in real-time intrusion detection. Specifically, in off-line analysis, it is implicitly assumed that all connections have already finished; therefore, we have

> the luxury to compute all the features and check the detection rules one by one. Whereas in real-time detection, we need to detect and respond to an intrusion as soon as it happens, often during an ongoing connection. We are therefore studying how to generate an efficient real-time execution plan for a detection ruleset.

(Wenke Lee and Salvatore J. Stolfo, *A framework for constructing features and models for intrusion detection systems*, ACM Transactions on Information and System Security (TISSEC), Vol. 3, Issue 4, 227–261 (November 2000)) (emphasis added). This definition is further supported by a 1998 academic survey on Internet Security defining "real time" as "constantly updated" noting:

> [m]isuse and anomaly detectors often run in real time, constantly checking a network or system — a file server, Web server, or perhaps a database or Notes server — for patterns of misuse or other inconsistencies. For example, they can be set up to watch over a Web server to ensure that key Web pages are not modified, or they can be configured to oversee an internal private network for unauthorized or never before-seen traffic.

(Frederick M. Avolio, *"Putting it together a multi-dimensional approach to Internet security,"* netWorker, Vol. 2, Issue 2, 15-22 (April/May 1998)) (emphasis added). It is important to note that systems categorized as "real-time" according to this definition do not have any precise timing constraints. Consistent with this definition, there are no mentions of precise timing in Douglas.

18. The term "hard real-time" is always used in a more narrow sense of "deterministic" or "satisfying precise time constraints" or "meeting deadlines" as shown by the following:

> [a] system is said to be real-time if the total correctness of an operation depends not only upon its logical correctness, but also upon the time in which it is performed. The classical conception is that in a hard or immediate real-time system, the completion of an operation after its deadline is considered useless -

> ultimately, this may lead to a critical failure of the complete system. A soft real-time system on the other hand will tolerate such lateness, and may respond with decreased service quality (e.g., dropping frames while displaying a video).

(http://en.wikipedia.org/wiki/Real-time#Hard_and_soft_real-time_systems).

This definition of "hard real-time" is well established in the technical literature. For example, a publication from 1991 is consistent with this definition stating:

> [t]here are many computer applications in which computations must satisfy stringent timing constraints, that is one must guarantee that those computations must be completed before specified deadlines. ... Such computer systems are called "hard-real-time" systems".

(Jia Xu and David Lorge Parnas, *On satisfying timing constraints in hard-real-time systems*, ACM SIGSOFT Software Engineering Notes, Vol. 16, Issue 5, 132-146 (December 1991)). This definition is further supported by the widely cited tutorial "Hard Real-Time Systems." (John A. Stankovic and Krithi Ramamritham, Hard Real-Time Systems (1988)) (note that both of these editors were on the doctoral committee of the inventor of the present invention).

19. With respect to claim 2, the Examiner asserts:

> [a]s to claim 2, Douglas discloses in a computer system running a real-time operating system.

(December 14, 2007 Office Action at p. 5). But Douglas makes no mention of a real-time operating system and more particularly, Douglas never refers to a "hard real-time operating system" that can assure timing of an application. In fact, Douglas specifically only refers to Linux and MS Windows, neither of which are hard real-time in its usual form.

20. With respect to claim 27 the Examiner asserts:

> Douglas discloses wherein the first real-time thread is further configured to check a set of integrity markers of the non-real-time kernel (column 2, lines 45-50).

(December 14, 2007 Office Action at p. 9). However, column 2, lines 45-50 of Douglas have no reference to threads, to real-time, or to one kernel checking another kernel. Instead, Douglas in Col 9, lines 45-50 refers to an application reading logs produced by the operating system hosting the application.

21. The Examiner makes the same error again in rejecting claim 31:

> Douglas discloses wherein the second real-time thread is further configured to check a set of integrity markers of the real-time kernel.

(December 14, 2007 Office Action at p. 9). As described above, Douglas makes no reference to any threads at all.

22. The novel aspects of the present invention rest in the application of "hard real-time" techniques to intrusion detection and prevention. All the claims of the application depend on the ability of the hard real-time operating system to guarantee that software will meet stringent timing constraints. Douglas in no way discloses, suggests, or even comes close to teaching such techniques. The Examiner writes that Douglas discloses:

> an external monitor connected to the network (col 2, lines 30-50), wherein the security process is configured to periodically, in hard real-time, check the integrity of the application and/or a data element used by the application.

(Col. 9, Lines 3-15) (emphasis added). Douglas instead discloses a continuous update of the integrity checks – no hard real-time at all. The novelty of the present invention is

precisely in this difference. On-line integrity checks are well established much before Douglas. As is disclosed in the application for the present invention, the utility of a hard real-time integrity check is that attackers are faced with the burden of completing an attack before a deadline expires.

23. In my opinion, since the distinguishing feature of the present invention is its use of hard real-time scheduling and event response capabilities and nothing in Douglas depends on such facilities, the difference should be clear. The application is significantly different and an improvement over well known techniques of security monitors, of which Douglas is an example. The present application is applying hard real-time techniques in a novel way to create new security features.

24. Accordingly, it is my opinion that one having ordinary skill in the art, at the time that the present application was filed, would not consider Douglas to be suggestive of the features of the present application for which the Examiner cites to Douglas.

B. The O'Neal Patent

25. O'Neal discloses a system and method for providing message and voice telephone service through the Internet that interfaces with the ordinary telephone network.

26. The Examiner cites to O'Neal as showing a deterministic network noting:

> O'Neal discloses a deterministic network (O'Neal teaches deterministic network by setting a predetermined time on response messages; responds [sic] are sent within a predetermined amount of time which makes the network to be deterministic [sic] – e.g.see col. 19, lines 10-20)

(December 14, 2007 Office Action at p. 4). However, nowhere in O'Neal is the term "deterministic" used, nor is any reference made to any timing properties of the network that might be construed as deterministic. In the portion cited by the Examiner (Col. 19, Lines 10-20), O'Neal discusses a means for detecting communication failures for devices over the Internet and/or T1 networks. These are not deterministic networks as the term is commonly used in reference to hard real-time systems.

27. O'Neal, column 19, describes a monitor that sends requests through a hub and a firewall to software systems such as web servers. None of these components offer any timing guarantees. Standard network protocols used for communication with those threads are TCP and UDP, neither of which has any underlying timing guarantees. As such, these protocols cannot be used to implement a time constrained method as claimed in the present application. The response times referenced in O'Neal, column 19, are periods of multiple seconds which are long enough so that failure to respond corresponds with a high probability of network or component failure. Such time periods are found in many networked systems that are not usually considered hard real-time. For example, the TCP protocol includes a 2 second time out. The challenge/response method claimed in the present application cannot be implemented by O'Neal's monitor precisely because the network and the components on the network offer no timing guarantees.

28. The Examiner also cites to O'Neal for the feature of transmitting a response to an external monitor within five milliseconds. The Examiner states:

> O'Neal discloses wherein the challenge handler is configured to provide a response within about one millisecond (col. 9, lines 10-25)

(December 14, 2007 Office Action at p. 4-5). There is no disclosure of timing at all in O'Neal, and the cited discussion in Column 9 describes users typing passwords and other completely non-deterministic and non-real-time activities that are not feasible within 1 millisecond. In fact, the word "millisecond" does not appear in O'Neal and the word "second" is only used to describe order, not as a unit of time.

29. Accordingly, it is my opinion that one having ordinary skill in the art, at the time that the present application was filed, would not consider O'Neal to be suggestive of the features of the present application for which the Examiner cites to O'Neal.

C. The REDSonic Reference

30. The Examiner cites REDSonic's version of a "dual-kernel" operating system as teaching the dual-kernel operating system of claims 3, 27, and 31 noting:

> it would be obvious to one of ordinary skill in the art at the time of the invention was made to modify the teachings of Sonic as taught by Douglas in order to provide notification of the intrusion or intrusion attempts.

(December 14, 2007 Office Action at p. 9).

31. REDSonic is not directed to a security apparatus at all.

32. While REDSonic discloses a dual-kernal operating system, the dual-kernel operating system was patented by the present inventor in 1998 (the REDSonic reference is dated 2002) and there are no examples in the literature or in disclosed patents combining that invention or other multi-kernel operating systems such as Hypervisors with intrusion detection, challenge/response, or any of the other components of the invention disclosed here. The MERT (Multi-Environment Real-Time)

system had a real-time operating system running on top of a non-real-time operating system and it was described in the literature in 1975. In all that time, there has been a large body of literature published on intrusion detection and many patent applications filed and patents granted. To the inventor's knowledge, prior to the filing of the present application, methods of using hard real-time capabilities of such systems for intrusion detection or any other security functionality have not been previously disclosed. Despite the fact that security and intrusion detection have been highly studied and commercialized fields for many years, the fact that no one has previously disclosed using hard real-time techniques is indicative of the non-obviousness of the invention described in the present application.

33. The combination of REDSonic with Douglas does not yield all of the features of claims 3, 27, and 31. As discussed above, Douglas fails to disclose or suggest threads being utilized to monitor the integrity of an application as required by the claims.

V. Conclusion

34. In my opinion, none of the cited prior art describes security features provided through hard real-time techniques, as described and claimed in the present application. Further, it is my opinion that, at the time of the invention, one skilled in the art would not have consulted the references cited by the Examiner and, even if consulted, would not have been motivated to use the disclosures because of their above-described deficiencies and differences.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the '469 application or any patent issued thereon.

Date: 5/14/08

Victor J. Yodaiken